

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS:

1. (currently amended) A method for serially aligning database transactions comprising at least two databases coupled to their associated database management systems, comprising the steps of: ~~in which~~

[[ - the]] initiating a first transaction ~~is initiated~~ in the first database[[ ,]];

[[ -]] linking at least one transaction trigger including attributes ~~is linked~~ into said first transaction[[ ,]];

[[ -]] ending said first transaction ~~is ended~~ in the first database[[ ,]];

~~is characterized in that, it further comprises step, in which~~  
firing at least one said trigger ~~is fired~~ in at least one first database; and

immediately after the ending and firing steps are completed,  
initiating at least one second transaction ~~is initiated~~ in the first database to invoke a remote database ~~operation~~ transaction in at least one second database according to at least some of the attributes in the trigger.

2. (currently amended) A method according to claim 1, ~~characterized in that~~ wherein the said trigger is a deferred database operation defined for at least one data manipulation operation.

3. (currently amended) A method according to claim 1, ~~characterized in that~~ wherein the execution of the second transaction is blocked until the said trigger fires.

4. (currently amended) A method according to claim 1, ~~characterized in that~~, wherein a database system comprises at least one master database and at least one replica database, and the data synchronization between the master and replica databases is master-initiated.

5. (currently amended) A method according to claim 1, ~~characterized in that~~ wherein the transactionally consistent set of data in a database comprises configuration data.

6. (currently amended) A method according to claim 5, ~~characterized in that~~ wherein the device changes its configuration to reflect the changed data right after the data has committed in the database.

7. (currently amended) A method according to claim 1, ~~characterized in that~~ wherein the related software processes, like other database server or a client application, are informed about transactional changes by the data management server.

8. (currently amended) A method according to claim 1, ~~characterized in that~~ wherein the method executes tasks and operations in a database transaction context.

9. (currently amended) A method according to claim 1, ~~characterized in that,~~ wherein any of the said transactions are executed in separate database connections or in a shared connection with another said transaction or another transaction.

10. (currently amended) A method according to claim 1, ~~characterized in that~~ wherein the method is compatible with at least one of the following communication specifications: TCP/IP, CDMA, GSM, HSCSD, GPRS, WCDMA, EDGE, UMTS, Bluetooth, Teldesic, Iridium, Inmarsat, WLAN, DIGI-TV and imode.

11. (currently amended) A method according to claim 1, ~~characterized in that~~ wherein the method is compatible with at least one of the following operating systems and is used in at least one terminal including an application, replica database of the database system Unix, MS-Windows, EPOC, NT, MSCE, Linux, PalmOS, GEOS, VxWorks, Pocket PC and any upgrade of these.

12. (currently amended) A method according to claim 1, ~~characterized in that~~ wherein at least one of the following operating systems is used in at least one server including an application master database of the database system: Unix, MS-Windows, VxWorks, NT and Linux and any upgrade of these.

13. (currently amended) A method for serially aligning database transactions comprising at least two databases coupled to their associated database management systems, comprising steps of: ~~in which~~

[[~~-~~]] initiating the first transaction ~~is initiated~~ in the first database[[,]];

[[~~-~~]] linking at least one transaction trigger including attributes ~~is linked~~ into said first transaction[[,]];

[[~~-~~]] ending said first transaction ~~is ended~~ in the first database[[,]];

~~is characterized in that, it further comprises step, in which~~  
firing at least one said trigger ~~is fired~~ in at least one first database; and

immediately after the ending and firing steps are completed,  
initiating at least one second transaction ~~is initiated~~ in the  
first database to synchronize data in at least one second database from at least one first database according to at least some of the attributes in the trigger.

14. (currently amended) A method according to claim 13, ~~characterized in that~~ wherein the set of data of the second transaction comprises data for performing push-style or push-pull-style synchronization.

15. (currently amended) A method according to claim 13, ~~characterized in that~~ wherein the said trigger is a procedure call defined for at least one data manipulation operations.

16. (currently amended) A method according to claim 13, ~~characterized in that~~ wherein the execution of the second transaction is blocked until the said trigger fires.

17. (currently amended) A method according to claim 13, ~~characterized in that,~~ wherein a database system comprises at least one master database and at least one replica database, the push synchronization data between the master and replica databases is master-initiated and pull synchronization data between the master and replica databases is replica-requested.

18. (currently amended) A method according to claim 13, ~~characterized in that~~ wherein the transactionally consistent set of data in a database comprises system configuration data.

19. (currently amended) A method according to claim 18, ~~characterized in that~~ wherein the device changes its configuration to reflect the changed data right after the data has committed in the database.

20. (currently amended) A method according to claim 13, ~~characterized in that~~ wherein the related software processes,

like other database server or a client application, are informed about transactional changes by the data management server.

21. (currently amended) A method according to claim 13, ~~characterized in that~~ wherein the method executes tasks and operations in a database transaction context.

22. (currently amended) A method according to claim 13, ~~characterized in that~~ wherein the method is compatible with at least one of the following communication specifications: TCP/IP, CDMA, GSM, HSCSD, GPRS, WCDMA, EDGE, UMTS, Bluetooth, Teldesic, Iridium, Inmarsat, WLAN, DIGI-TV and imode.

23. (currently amended) A method according to claim 13, ~~characterized in that~~ wherein the method is compatible with at least one of the following operating systems and is used in at least one terminal including an application, replica database of the database system Unix, MS-Windows, EPOC, NT, MSCE, Linux, PalmOS, GEOS, VxWorks, Pocket PC and any upgrade of these.

24. (currently amended) A method according to claim 13, ~~characterized in that~~ wherein at least one of the following operating systems is used in at least one server including an application master database of the database system: Unix, MS-Windows, VxWorks, NT and Linux and any upgrade of these.

25. (currently amended) A method according to claim 13, ~~characterized in that,~~ wherein any of the said transactions are

executed in separate database connections or in a shared connection with another said transaction or another transaction.

26. (currently amended) An arrangement for serially aligning database transactions comprising:

at least two databases and the associated database management system,

~~comprising~~ means for linking at least one transaction trigger including attributes into the first transaction in the first database, and ~~characterized in that, the arrangement further comprises~~

means for initiating at least one second transaction to ~~push data into~~ invoke a remote database transaction in at least one second database according to at least some of the attributes in the trigger immediately after said first transaction is ended in the first database and ~~thereafter~~ said trigger is fired in at least one first database.

27. (currently amended) An arrangement according to claim 26, ~~characterized in that it comprises~~ comprising at least one master database and one replica database coupled to associated database management systems.

28. (currently amended) An arrangement according to claim 26, ~~characterized in that~~ wherein the transactionally consistent set of data in a database comprises system configuration data.

29. (currently amended) An arrangement according to claim 26, ~~characterized in that~~ wherein at least the second database can be part of a router coupled to the application.

30. (currently amended) An arrangement according to claim 26, ~~characterized in that~~ further comprising a storage medium that is one of a memory [[and/or]] and a disk.

31. (currently amended) An arrangement according to claim 26, ~~characterized in that~~ wherein the arrangement and/or database system is compatible with at least one of the following communication specifications: TCP/IP, CDMA, GSM, HSCSD, GPRS, WCDMA, EDGE, UMTS, Bluetooth, Teldesic, Iridium, Inmarsat, WLAN, DIGI-TV and imode.

32. (currently amended) An arrangement according to claim 26, ~~characterized in that~~ wherein at least one of the following operating systems and is used in at least one terminal including an application, replica database of the database system Unix, MS-Windows, EPOC, NT, MSCE, Linux, PalmOS, GEOS, VxWorks, Pocket PC and any upgrade of these.

33. (currently amended) An arrangement according to claim 26, ~~characterized in that~~ wherein at least one of the following



operating systems is used in at least one server including an application master database server and/or the schema management node: Unix, MS-Windows, VxWorks, NT and Linux and any upgrade of these.

34. An arrangement for serially aligning database transactions comprising:

at least two databases and the associated database management system, comprising means for linking at least one transaction trigger including attributes into the first transaction in the first database, and ~~characterized in that, the arrangement further comprises~~

means for initiating at least one second transaction in the first database to synchronize data in at least one second database from at least one first database according to at least some of the attributes in the trigger immediately after said first transaction is ended in the first database and ~~thereafter~~ said trigger is fired in at least one first database.

35. (currently amended) An arrangement according to claim 34, ~~characterized in that~~ wherein the set of data comprises data of push or push-pull synchronization.

36. (currently amended) An arrangement according to claim 34, ~~characterized in that it comprises~~ comprising at least one

master database and one replica database coupled to associated database management system.

37. (currently amended) An arrangement according to claim 34, ~~characterized in that~~ wherein the transactionally consistent set of data in a database comprises system configuration data.

38. (currently amended) An arrangement according to claim 34, ~~characterized in that~~ wherein at least the second database ~~can be~~ is part of a router coupled to the application.

39. (currently amended) An arrangement according to claim 34, ~~characterized in that~~ further comprising a storage medium that is one of a memory [[and/or]] and a disk.

40. (currently amended) An arrangement according to claim 34, ~~characterized in that~~ wherein the arrangement and/or the database system is compatible with at least one of the following communication specifications: TCP/IP, CDMA, GSM, HSCSD, GPRS, WCDMA, EDGE, UMTS, Bluetooth, Teldesic, Iridium, Inmarsat, WLAN, DIGI-TV and imode.

41. (currently amended) An arrangement according to claim 34, ~~characterized in that~~ wherein at least one of the following operating systems is used in at least one terminal including an application, replica database of the database system Unix, MS-Windows, EPOC, NT, MSCE, Linux, PalmOS, GEOS, VxWorks, Pocket PC and any upgrade of these.

42. (currently amended) An arrangement according to claim 34, ~~characterized in that~~ wherein at least one of the following operating systems is used in at least one server including an application master database server and/or the schema management node: Unix, MS-Windows, VxWorks, NT and Linux and any upgrade of these.